



# National Textile University

## Department of Computer Science

Subject:

Operating System

---

Submitted to:

Sir Nasir Mehmood

---

Submitted by:

Eman Babar

---

Reg. number:

23-NTU-CS-FL-1148

---

Semester:

5<sup>th</sup>- A

---

# LAB-3-Home-Task

## Part 1: File and Directory Operations

1. Create the following directory structure in your home directory:

```
Lab_3/  
├── docs/  
│   └── drafts/  
├── data/  
│   ├── raw/  
│   └── processed/  
└── scripts/
```

```
ohail_arwar@SALMAN:~$ mkdir -p Lab_3/docs/drafts  
ohail_arwar@SALMAN:~$ mkdir -p Lab_3/data/raw  
ohail_arwar@SALMAN:~$ mkdir -p Lab_3/data/processed  
ohail_arwar@SALMAN:~$ mkdir -p Lab_3/scripts  
ohail_arwar@SALMAN:~$
```

2. Inside docs/:

- Create three files: intro.txt and notes.txt and summary.txt
- Add at least two lines of text into each using echo >>.
- Copy summary.txt into the drafts/ folder using **cp** command.

```
ohail_arwar@SALMAN:~/Lab_3/docs$ echo "This is second line in intro file." >> intro.txt
ohail_arwar@SALMAN:~/Lab_3/docs$ echo "These are notes." >> notes.txt
ohail_arwar@SALMAN:~/Lab_3/docs$ echo "This is second line in notes." >> notes.txt
ohail_arwar@SALMAN:~/Lab_3/docs$ echo "Summary starts here." >> summary.txt
ohail_arwar@SALMAN:~/Lab_3/docs$ echo "This is second line n summary." >> summary.txt
ohail_arwar@SALMAN:~/Lab_3/docs$ cp summary.txt drafts/
ohail_arwar@SALMAN:~/Lab_3/docs$
```

### 3. Inside data/raw/:

- Create two files: raw1.txt, raw2.txt
- Append the current date into cp command. raw1.txt using the date command.
- Move raw2.txt into processed/ using mv source destination mv. The syntax is:  
**mv source destination**

```
ohail_arwar@SALMAN:~/Lab_3/docs$ cd ~/Lab_3/data/raw
ohail_arwar@SALMAN:~/Lab_3/data/raw$ echo "Raw file 1" >> raw1.txt
ohail_arwar@SALMAN:~/Lab_3/data/raw$ echo "Raw file 2" >> raw2.txt
ohail_arwar@SALMAN:~/Lab_3/data/raw$ date >> raw1.txt
ohail_arwar@SALMAN:~/Lab_3/data/raw$ mv raw2.txt ../processed/
ohail_arwar@SALMAN:~/Lab_3/data/raw$
```

### 4. Inside scripts/:

- Create a script named **hello.sh** with the following content:  
**echo "Hello World"**  
**pwd**  
**ls -lh**
- Later, you will make it executable (in Part 3).

```
ohail_arwar@SALMAN: ~/Lab_3/scripts
ohail_arwar@SALMAN:~/Lab_3/data/raw$ cd ~/Lab_3/scripts
ohail_arwar@SALMAN:~/Lab_3/scripts$ echo 'echo "Hello World"' >> hello.sh
ohail_arwar@SALMAN:~/Lab_3/scripts$ echo 'pwd' >> hello.sh
ohail_arwar@SALMAN:~/Lab_3/scripts$ echo 'ls -lh' >> hello.sh
ohail_arwar@SALMAN:~/Lab_3/scripts$
```

5. Display the directory structure recursively and take a screenshot:

**Ls -R**

```
ohail_arwar@SALMAN ~/Lab_3
./data/raw:
raw1.txt

./docs:
drafts  intro.txt  notes.txt  summary.txt

./docs/drafts:
summary.txt

./scripts:
hello.sh
ohail_arwar@SALMAN:~/Lab_3$
```

## **Part 2: Practice with Basic Linux Commands**

Run the following commands inside Lab\_3/ and note their outputs:

- pwd → Show current working directory.
- whoami → Display the current logged-in user.
- touch extra.txt → Create an empty file.
- cat intro.txt → Display file contents.
- rm extra.txt → Delete a file.
- history | tail -n 5 → Show your last 5 executed commands.

```
ohail_arwar@SALMAN:~/Lab_3$ cd ~/Lab_3/docs
ohail_arwar@SALMAN:~/Lab_3/docs$ pwd
/home/ohail_arwar/Lab_3/docs
ohail_arwar@SALMAN:~/Lab_3/docs$ whoami
ohail_arwar
ohail_arwar@SALMAN:~/Lab_3/docs$ touch extra.txt
ohail_arwar@SALMAN:~/Lab_3/docs$ cat intro.txt
This is an intro file.
This is second line in intro file.
ohail_arwar@SALMAN:~/Lab_3/docs$ rm extra.txt
ohail_arwar@SALMAN:~/Lab_3/docs$ history | tail -n 5
363 whoami
364 touch extra.txt
365 cat intro.txt
366 rm extra.txt
367 history | tail -n 5
ohail_arwar@SALMAN:~/Lab_3/docs$
```

- clear → Clear the terminal. Take screenshots of commands and outputs

```
ohail_arwar@SALMAN:~/Lab_3/docs$
```

### **Part 3: File Permissions and Ownership**

1. Change the permissions of hello.sh so that:
  - Owner → Read, Write & Execute
  - Group → Read, Write & Execute
  - Others → No permissions
  - Run the script using:  
./hello.sh
2. Change the permissions of intro.txt using numeric notation so that:
  - Owner → Read & Write

- Group → Read & Write
  - Others → Read only 3.
3. Change the permissions of **notes.txt** using **symbolic notations** so that **others** have any permission on it.
  4. Verify all changes with:

ls-l

```
ohail_arwar@SALMAN:~/Lab_3/docs$ cd ~/Lab_3/scripts
ohail_arwar@SALMAN:~/Lab_3/scripts$ chmod 770 hello.sh
ohail_arwar@SALMAN:~/Lab_3/scripts$ ./hello.sh
Hello World
/home/ohail_arwar/Lab_3/scripts
total 4.0K
-rwxrwx--- 1 ohail_arwar ohail_arwar 30 Oct 7 17:27 hello.sh
ohail_arwar@SALMAN:~/Lab_3/scripts$ cd ~/Lab_3/docs
ohail_arwar@SALMAN:~/Lab_3/docs$ chmod 664 intro.txt
ohail_arwar@SALMAN:~/Lab_3/docs$ chmod o-rwx notes.txt
ohail_arwar@SALMAN:~/Lab_3/docs$ ls -l
total 16
drwxr-xr-x 2 ohail_arwar ohail_arwar 4096 Oct 7 17:17 drafts
-rw-rw-r-- 1 ohail_arwar ohail_arwar 58 Oct 7 17:15 intro.txt
-rw-r----- 1 ohail_arwar ohail_arwar 47 Oct 7 17:15 notes.txt
-rw-r--r-- 1 ohail_arwar ohail_arwar 52 Oct 7 17:16 summary.txt
ohail_arwar@SALMAN:~/Lab_3/docs$
```

## Part 4: Reading and Searching Files

Inside docs/:

1. Count the number of lines, words, and characters in **notes.txt** using **wc**
2. Show only the first 2 lines of **summary.txt** using **head -n 2**
3. Show the last line of **summary.txt** using **tail -n 1**
4. Search for a keyword (of your choice) in **intro.txt** using **grep**.

```
ohail_arwar@SALMAN:~/Lab_3/docs$ wc notes.txt
 2  9 47 notes.txt
ohail_arwar@SALMAN:~/Lab_3/docs$ head -n 2 summary.txt
Summary starts here.
This is second line n summary.
ohail_arwar@SALMAN:~/Lab_3/docs$ tail -n 1 summary.txt
This is second line n summary.
ohail_arwar@SALMAN:~/Lab_3/docs$ grep "Summary" intro.txt
ohail_arwar@SALMAN:~/Lab_3/docs$ grep "intro" intro.txt
This is an intro file.
This is second line in intro file.
ohail_arwar@SALMAN:~/Lab_3/docs$
```

## Part 5: Linux Process Commands

### 1. Exploring Processes

- Use **ps -ef** and identify 3 processes running on your system. Note their PID, PPID, and command.
- Run **top** for 20–30 seconds. Write down:
  - Which process is consuming the most CPU
  - Which process is consuming the most memory

```
ohail_arwar@SALMAN: ~/Lab_3/docs$ ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1        0   0  0:00 ?        00:00:19 /sbin/init
root           2        1   0  0:00 ?        00:00:00 /init
root           6        2   0  0:00 ?        00:00:03 plan9 --control-socket 7 --log-level 4 --server-fd 8 -root      78      1  4  1
7:01 ?       00:00:17 /usr/lib/systemd/systemd-journald
root          133        1   0  0:00 ?        00:00:01 /usr/lib/systemd/systemd-udevd
systemd+      187        1   0  0:00 ?        00:00:01 /usr/lib/systemd/systemd-resolved
systemd+     188        1   0  0:00 ?        00:00:02 /usr/lib/systemd/systemd-timesyncd
root          197        1   0  0:00 ?        00:00:00 /usr/sbin/cron -f -P
message+     198        1   0  0:00 ?        00:00:12 @dbus-daemon --system --address=systemd: --nofork --noroot      214      1  0  1
7:01 ?       00:00:05 /usr/lib/systemd/systemd-logind
root          216        1   0  0:00 ?        00:00:01 /usr/libexec/wsl-pro-service -vv
root          219        1   0  0:00 ?        00:00:00 /sbin/agetty -o -p -- \u --noclear --keep-baud - 11520syslog      231      1  0  1
7:01 ?       00:00:00 /usr/sbin/rsyslogd -n -iNONE
root          234        1   0  0:00 ?        00:00:00 /sbin/agetty -o -p -- \u --noclear - linux
root          240        1   0  0:00 ?        00:00:00 /usr/bin/python3 /usr/share/unattended-upgrades/unattendedroot    328      2  0  1
7:01 ?       00:00:00 /init
root          329        1   0  0:00 ?        00:00:01 /init
ohail_a+     330        1   0  0:00 ?        00:00:01 pts/0    -bash
root          331        2   0  0:00 ?        00:00:00 /bin/login -f
ohail_a+     432        1   0  0:00 ?        00:00:01 /usr/lib/systemd/systemd --user
ohail_a+     439        1   0  0:00 ?        00:00:00 (sd-pam)
ohail_a+     484        1   0  0:00 ?        00:00:00 pts/1    -bash
ohail_a+     2401       330  99  18:08 pts/0    00:00:00 ps -ef
ohail_arwar@SALMAN: ~/Lab_3/docs$ top
top - 18:09:51 up 1:08, 1 user, load average: 0.46, 0.18, 0.13
Tasks: 23 total, 1 running, 22 sleeping, 0 stopped, 0 zombie
%Cpu(s):  0.0 us,  0.5 sy,  0.0 ni, 99.5 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem : 3877.2 total, 3064.4 free,  474.6 used,  479.0 buff/cache
MiB Swap: 1024.0 total, 1024.0 free,  0.0 used,  3402.7 avail Mem

   PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
    1 root        20   0  21860  12596  9396  S   0.0   0.3   0:19.95 systemd
    2 root        20   0  3072   1920  1792  S   0.0   0.0   0:00.11 init-systemd(+
    6 root        20   0  3120   2068  1920  S   0.0   0.1   0:03.26 init
   78 root       19  -1 66752 15496 14472  S   0.0   0.4   3:21.51 systemd-journ+
  133 root        20   0 25048   6144 4992  S   0.0   0.2   0:01.87 systemd-udevd
```

### 2. Practice with Infinite Process

- Start: **yes > /dev/null &**
- Locate its PID using **ps -ef | grep yes**
- Kill it using **kill <PID>** and verify using **ps**

```
ohail_arwar@SALMAN: ~/Lab_3/docs$ yes > /dev/null &
[1] 2755
ohail_arwar@SALMAN: ~/Lab_3/docs$ ps -ef | grep yes
ohail_a+  2755    2601  99 18:21 pts/0    00:00:12 yes
ohail_a+  2761    2601   0 18:21 pts/0    00:00:00 grep --color=auto yes
ohail_arwar@SALMAN: ~/Lab_3/docs$ kill 2755
ohail_arwar@SALMAN: ~/Lab_3/docs$ ps -ef | grep yes
ohail_a+  2775    2601   0 18:21 pts/0    00:00:00 grep --color=auto yes
[1]+  Terminated yes > /dev/null
ohail_arwar@SALMAN: ~/Lab_3/docs$
```

### 3. Foreground & Background Jobs



- Run **sleep 60** in **foreground** and terminate it with **Ctrl + C**.
- Run **sleep 60 &** in **background**, bring it to foreground with **fg**, stop using **Ctrl + Z**, then resume in background using **bg**.

```
ohail_arwar@SALMAN:~/Lab_3/docs$ sleep 60
^C
ohail_arwar@SALMAN:~/Lab_3/docs$ sleep 60 &
[1] 2821
ohail_arwar@SALMAN:~/Lab_3/docs$ fg
sleep 60
^Z
[1]+  Stopped                  sleep 60
ohail_arwar@SALMAN:~/Lab_3/docs$ bg
[1]+ sleep 60 &
ohail_arwar@SALMAN:~/Lab_3/docs$
```

## Part 6: C Programs on Processes

### Program 1 – Exec with top

- Modify the exec program so that the child runs **top** instead of **ls-l**.
- Run the program.
- In another terminal, use **ps -ef | grep top** (or run **top** ) to find the child's PID.
- Use the child's process ID to kill it manually.

The screenshot shows a Visual Studio Code editor with a C program in a file named `Q1.c`. The program uses `fork()` to create a child process. The child process calls `execlp("top", "top", NULL);` to execute the `top` command. The parent process calls `wait(NULL);` to wait for the child to finish. The output window shows the execution of the program, including the `top` command's output, which displays system statistics and a list of running processes. The parent process prints "Child finished" and returns 0.

```
1 #include <stdio.h>
2 #include <unistd.h>
3 #include <sys/wait.h>
4
5 int main() {
6     pid_t pid = fork();
7
8     if (pid == 0) {
9         execlp("top", "top", NULL);
10    } else {
11        wait(NULL);
12        printf("Child finished\n");
13    }
14    return 0;
15 }
```

Output:

```
top - 15:40:03 up 1 min, 1 user, load average: 1.47, 0.60, 0.22
%cpu(s): 5.4 us, 5.4 sy, 0.0 ni, 85.6 id, 3.0 wa, 0.0 hi, 0.6 si, 0.0 st
MiB Mem : 3877.2 total, 2179.6 free, 1188.0 used, 647.8 buff/cache
MiB Swap: 1024.0 total, 1024.0 free, 0.0 used, 2689.3 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
634 ohail_ar 20 0 52.5g 672580 55424 S 11.1 16.9 0:52.54 node
687 ohail_ar 20 0 1072988 72856 46720 S 8.5 1.8 0:04.09 node
63 root 19 -1 50356 15284 14388 S 3.6 0.4 0:04.10 systemd-journal
468 ohail_ar 20 0 11.3g 105900 50560 S 3.0 2.7 0:15.76 node
1 root 20 0 21704 12460 9260 S 0.7 0.3 0:06.63 systemd
880 ohail_ar 20 0 9292 5248 3200 R 0.7 0.1 0:00.14 top
205 message+ 20 0 9636 4864 4352 S 0.3 0.1 0:00.87 dbus-daemon
597 ohail_ar 20 0 947268 52096 41856 S 0.3 1.3 0:00.87 node
2 root 20 0 3072 1920 1792 S 0.0 0.0 0:00.12 init-systemd(ub
6 root 20 0 3072 1792 1792 S 0.0 0.0 0:00.00 init
121 root 20 0 25136 6016 4864 S 0.0 0.2 0:00.98 systemd-udev
184 systemd+ 20 0 21456 12288 10240 S 0.0 0.3 0:00.60 systemd-resolve
185 systemd+ 20 0 91024 7808 6912 S 0.0 0.2 0:00.33 systemd-timesyn
204 root 20 0 4236 2560 2432 S 0.0 0.1 0:00.11 cron

Child finished
ohail_arwar@SALMAN:~/05-1148/LAB-3-HomeTasks$
```

Second Terminal:



```
1 #include <stdio.h>
2 #include <unistd.h>
3 #include <sys/wait.h>
4
5 int main() {
6     pid_t pid = fork();
7
8     if (pid == 0) {
9         execlp("top", "top", NULL);
10    } else {
11        wait(NULL);
12        printf("Child finished\n");
13    }
14    return 0;
15 }
```

chail\_arwar@SALMAN:~/05-1148/LAB-3-HomeTasks\$ ps -ef | grep top

chail\_a+ 880 0 15:40 pts/4 00:00:00 top

chail\_a+ 989 0 15:40 pts/5 00:00:00 grep --color=auto top

chail\_arwar@SALMAN:~/05-1148/LAB-3-HomeTasks\$ kill 880

chail\_arwar@SALMAN:~/05-1148/LAB-3-HomeTasks\$ ls -l

Q1.c

chail\_arwar@SALMAN:~/05-1148/LAB-3-HomeTasks\$

## Program 2 – Incomplete Program

```
1 #include <stdio.h>
2 #include <unistd.h>
3 #include <sys/wait.h>
4
5 int main() {
6     pid_t pid = fork();
7
8     if (pid == 0) {
9         execlp("date", "date", NULL);
10    } else {
11        wait(NULL);
12        printf("Child finished\n");
13    }
14    return 0;
15 }
```

chail\_arwar@SALMAN:~/05-1148/LAB-3-HomeTasks\$ gcc Q2.c -o Q2-output

chail\_arwar@SALMAN:~/05-1148/LAB-3-HomeTasks\$ ./Q2-output

Thu Oct 9 15:40:50 PKT 2025

Child finished

chail\_arwar@SALMAN:~/05-1148/LAB-3-HomeTasks\$